逻辑与神经之间的桥

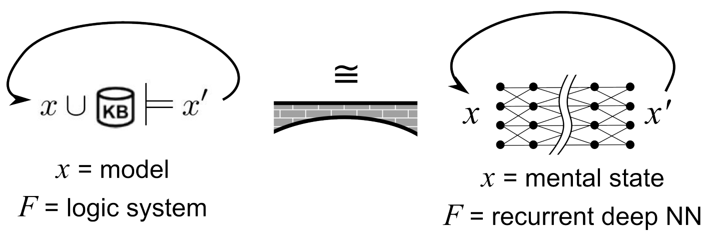
甄景贤 (King-Yin Yan)

General.Intelligence@Gmail.com

Abstract. Logic-based AI 和 connectionist AI 长久分裂，但笔者最近发现了可以统一两者的理论。

逻辑 AI 那边，「结构」很精细，但学习算法太慢；我的目的是建立一道「桥」，将逻辑 AI 的某部分结构**转移**到神经网络那边，这样可以融合两边的好处。

这个问题搞了很久都未能解决，因为逻辑 AI 那边的结构不是一般常见的数学结构，单是要要表述出来也有很大困难。直到我应用了 model theory 的观点，才找到满意的解决方法：



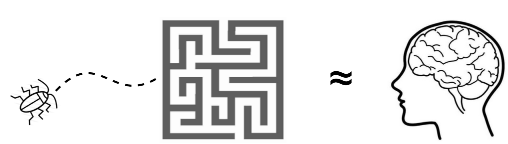
(1)

我首先解释 neural network 那边的结构，然后再解释 logic 那边的结构。

1 Neural architecture

Itamar Arel 在 2012 年 [1]、和我在 2016 年 [3]、独立地提出了同样的 cognitive architecture：合并 **增强学习**和**深度学习**。

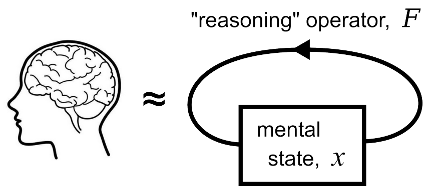
打个比喻来说，就是用强化学习去控制一隻智慧生物，在「思考空间」的迷宫中找最佳路径：



(2)

强化学习特别适合解决这类问题，可以参看我写的 tutorial。

关键是将「思考」看成是一个动态系统 (dynamical system)，它运行在思维状态 (mental states) 的空间中：



(3)

举例来说，一个思维状态可以是以下的一束命题：

- 我在我的房间内，正在写一篇论文。
- 我正在写一句句子的开头：「我在我的房间内，....」
- 我将会写一个动词词组 (verb phrase)：「正在写....」

思考的过程就是从一个思维状态 过渡 (transition) 到另一个思维状态。就算我现在说话，我的脑子也是靠思维状态记住我说话说到句子结构的哪部分，所以我才能组织句子的语法。

思维状态是一支向量 $\boldsymbol{x} \in X$，X 是全体思维空间，思考算子 (reasoning operator) $F : X \rightarrow X$ 是一个 endomorphism。

一个**动态系统 (dynamical system)** 可以用以下方法定义：

$$\text{离散时间：} \qquad \boldsymbol{x}_{n+1} = \boldsymbol{F}(\boldsymbol{x}_n) \tag{4}$$

$$\text{连续时间：} \qquad \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) \tag{5}$$

为方便起见，有时我会滥用 F 和 f 的表述（不区分连续和离散）。

一个（连续时间的）**控制系统 (control system)** 定义为：

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \tag{6}$$

其中 $\boldsymbol{u}(t)$ 是控制向量。控制论的目的就是找出最好的 $\boldsymbol{u}^*(t)$，令系统由初始状态 $\boldsymbol{x}_0$ 去到终点状态 $\boldsymbol{x}_\perp$。

动态规划 (dynamic programming) 的中心思想是 Bellman equation；我们根据 Bellman update 寻找状态空间中的最优路径。

注意：人工智能中的 **A* search**，是动态规划的一个特例。换句话说，用动态规划在某个空间中「漫游」，可以模拟 best-first 搜寻的功能。

我们的目标是学习 $F \in \{$ 无限维的算子空间 $\}$。实践上 F 可以用 deep learning network (dNN) 代表，换句话说 F 就是一个有很多 parameters 的非线性算子 (= 神经网络)。

一个**神经网络**基本上是：

$$F(\boldsymbol{x}) = \oint(W_1 \oint(W_2 ... \oint(W_L\ \boldsymbol{x}))) \tag{7}$$

其中 L 是层数，W 是每层的权重**矩阵**，$\oint$ 是对每个分量的 sigmoid function（其作用是赋予非线性）。

在这框架下，智能系统的运作可以分开成两方面：思考 和 学习。

思考即是根据已学得的知识（知识储存在 dNN 里），在思维空间中找寻 $\boldsymbol{x}$ 最优的轨迹，方法是用控制论计算 $\boldsymbol{u}^*$。$\boldsymbol{x}$ 的轨迹受 dNN 约束（系统只能依据「正确」的知识去思考），但思考时 dNN 是不变的。

学习就是学习神经网络 dNN 的 weights W。此时令 $u = 0$，即忽略控制论方面。

而很明显，「自由」的 F 算子没有「内部结构」，它能够学习的就像是甲由那样的、简单的「条件反射」行为。如果要达到人类的智慧，则要学习很久（到时我们都死了）。

所以问题就是要赋予 F 更多的**结构**，特别是逻辑结构。直观地说，越多的结构令**搜寻空间**越小，学习会越快。这是机器学习里面 inductive bias 的标准做法。

2 Logic-based AI

用数理逻辑 (mathematical logic) 模拟人的思想是可行的，例如有 deduction, abduction, induction 等这些模式，详细可见《Computational logic and human thinking》by Robert Kowalski, 2011. 这些方面不影响本文的阅读。值得一提的是，作者 Kowalski 是 logic programming，特别是 Prolog，的理论奠基人之一。

在经典逻辑 AI 中，「思考」是透过一些类似以下的步骤：

$$\text{前提} \vdash \text{结论} \tag{8}$$

$$\boxed{\text{今天早上下雨}} \vdash \boxed{\text{草地是湿的}} \tag{9}$$

亦即由一些命题(propositions) 推导到另一些命题。

推导必须依靠一些逻辑的法则命题 (rule propositions)，所谓「法则」是指命题里面带有 x 这样的变量(variables)：

$$\boxed{\text{地方 x 下雨}} \wedge \boxed{\text{x 是露天的}} \vdash \boxed{\text{地方 x 是湿的}} \tag{10}$$

这些法则好比「逻辑引擎」的燃料，没有燃料引擎是不能推动的。

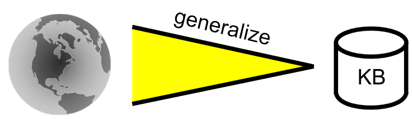
注意：命题里面的 x，好比是有「洞」的命题，它可以透过 substitution 代入一些实物 (objects)，而变成完整的命题。这种「句子内部」(sub-propositional) 的结构可以用 predicate logic （谓词逻辑）表达，但暂时不需要理会这些细节。

「所有人失恋了都会不开心」:

$$\forall z.\, \cancel{\heartsuit}(z) \rightarrow \frownie(z) \tag{11}$$

在数理逻辑中这算是一条 公理 (axiom)，但在 AI 中这些公理是从主体的经验中**学习**出来的，我们仍沿用「公理」这术语。在 AI 术语中，公理的集合叫 knowledge base，记作 KB。注意 KB 是一堆 **formulas** 的集合。

Logic-based AI 可以看成是将世界的「模型」压缩成一个 KB:



(12)

世界模型是由大量的逻辑式子经过组合而生成的，有点像向量空间是由其「基底」生成；但这生成过程在逻辑中特别复杂，所以符号逻辑具有很高的压缩比，但要学习一套逻辑 KB，则相应地也有极高的复杂度。

3 逻辑的结构

一个逻辑系统可以这样定义：

- 一些 constant symbols，predicate symbols，和 function symbols
- 由上述的原子建立 命题 (propositions)
- 命题之间可以有连接词：$\neg, \wedge, \vee$ 等
- 建立 逻辑后果 (consequence) 关系：$\Gamma \vdash \Delta$

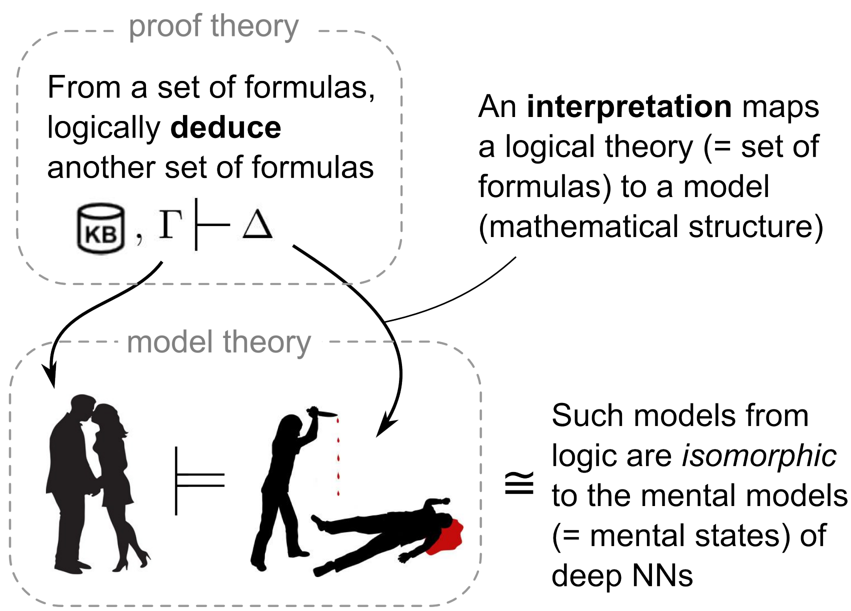
我个人认为 relation algebra 比较接近人类自然语言，但在数理逻辑研究中最通用的逻辑是 first-order logic (FOL)。然而这并不是重点，因为各种逻辑基本上是等效的，而且相互之间可以很容易地转换。以下集中讨论 FOL。

我以前花了很多时间思考怎样将逻辑的 $\vdash$ 关系过渡到神经网络去，但发觉这个目标非常 elusive。

在认知科学里，有很多人相信大脑的内部的 representation 是一些所谓 “mental models”，而很少人会相信大脑使用一些像命题那样的符号结构做 representation，甚至用 λ-calculus 那样的符号 manipulation 去思考。

另一方面，逻辑是几百年来发展起来的关於人类思考的规律；逻辑的描述是正确的；逻辑和神经之间必然有一个 correspondence，因为它们都在做同样的事：智能。

所以我终於发现到，logic-neuro correspondence 必须透过 model theory [?] [?] 才能达成：



(13)

4 Model theory

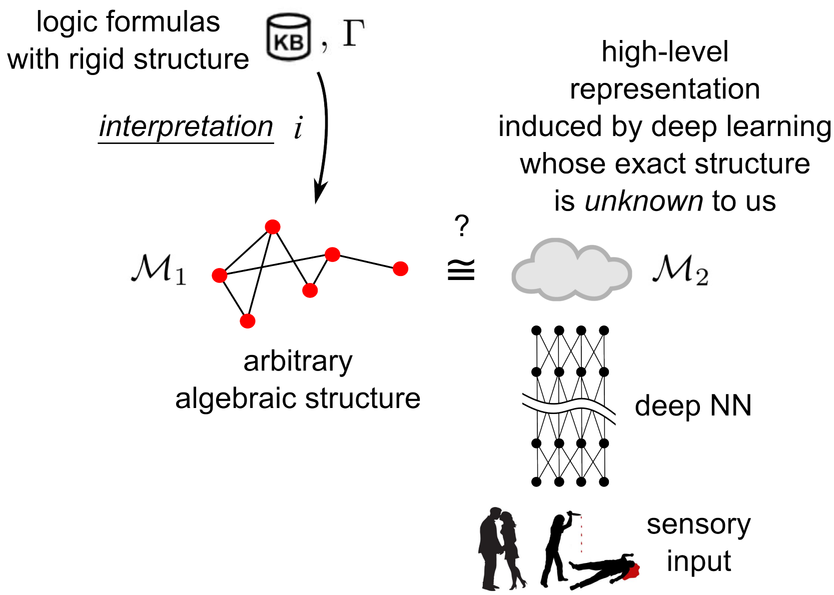
如果用範畴论的方法表示：

$$
\begin{tikzcd}
\mathcal{L} \arrow[d, "i"] & & \\
\mathcal{M}_1 & \simeq & \mathcal{M}_2 \\
& & \mathcal{S} \arrow[u, "\text{dNN}"']
\end{tikzcd}
\tag{14}
$$

- $\mathcal{L}$ = category of logic theories (= sets of formulas)
- i = interpretation maps
- $\mathcal{M}_1$ = category of models (from logic)
- $\mathcal{M}_2$ = category of models (from deep NNs)

- $\mathcal{S}$ = sensory input

上图等同於下面的卡通解释：



(15)

换句话说，$\mathcal{M}_2 =$  是由深度学习 induce 出来的结构；但它的结构对我们来说是不透明的（这是神经网络的弱点）。

而 $\mathcal{M}_1 =$  的结构是 free 的；换句话说，那 i map 的 source domain 是固定的，但 target domain 是自由的。这导致 i map 的学习很困难，因为 $\mathcal{M}_1$ 和 $\mathcal{M}_2$ 的结构都不清楚。必须更详细分析 $\mathcal{M}_1, \mathcal{M}_2$ 的结构。

5 Model 和 interpretation 的结构

在模型论中，$\mathcal{L}$ 是逻辑句子的範畴，$\mathcal{M}_1 =$  可以是任何抽象代数结构。只需把 $\mathcal{L}$ 中的 constants, predicates, relations, functions 映射到 $\mathcal{M}_1$ 就行。为简化讨论，我们只考虑 constants 和 relations，因为二者是逻辑中最**本质**的东西。

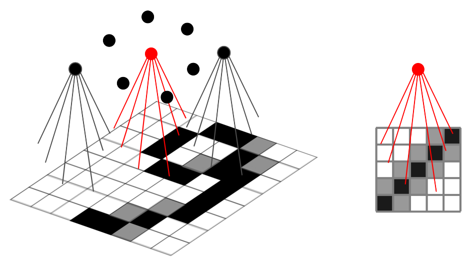
$\mathcal{L}$	$\xrightarrow{i}$	$\mathcal{M}_1$
constant symbol	$\overset{i}{\mapsto}$	
relation symbol	$\overset{i}{\mapsto}$	

(16)

问题是在神经那边缺乏  的结构。一直以来，人们习惯把神经网络看成是 "black box"，但如果我们不知道 $\mathcal{M}_2 =$  的结构，就无法建立 $\mathcal{M}_1 \simeq \mathcal{M}_2$ 的 isomorphism。

那么，神经网络的 representation 究竟是什么结构？

考虑最简单的情况，例如提取 digit "9" 的特徵的一层网络。这层网络可以有很多神经元（左图），每个神经元局部地覆盖输入层，即所谓视觉神经元的 local receptive field（右图）。



(17)

假设红色的神经元专门负责辨识「对角线」这一特徵。它的方程式是 $\boldsymbol{y} =$ $(W\boldsymbol{x})$。矩阵 W 的作用是 affine「旋转」特徵空间，令我们想要的特徵集中在某一方向。然后再用 「**扯开**」想要的特徵和不想要的特徵[1]。Sigmoid 之后的输出，代表某类特徵的**存在与否**。

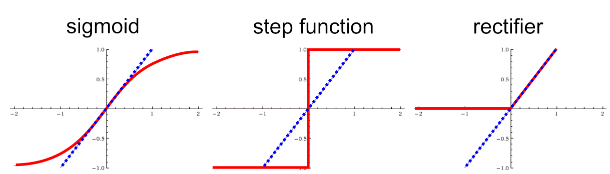
换句话说：每个神经元的输出其实代表某个 feature 的存在与否。
而，更高层的神经元代表下层 features 之间的**关系**。

凭这个思路推广，可以推测这样的 correspondence：

$$\begin{array}{rccl} & \mathcal{M}_1 & \simeq & \mathcal{M}_2 \\ \text{constant} & \bullet & \Leftrightarrow & \text{neuron} \\ \text{relation} & \bullet\!\!-\!\!\bullet & \Leftrightarrow & \text{relation between higher and lower neurons} \end{array} \tag{19}$$

但要注意的是这对应未必是一对一的，可能是一个 constant 对应几个 neurons 的**线性组合**。

[1] 所谓「扯」(stretch) 的意思是说，将本来邻近的两点的距离非线性地拉远。看看以下各种常见的激活函数，它们全都是相对於 identity $y = x$ 的非线性 distortion：



(18)

这和 Steven Smale 提出的「马蹄」[2] 非常类似，它是制造混沌的处方之一。换句话说，「拉扯」然后放回原空间，如此不断重复，就会产生混沌 [?] [?]。其作用类似於「搓面粉」，所以另一个变种也叫做 baker map。

References

1. Itamar Arel. *Deep reinforcement learning as Foundations for Artificial Intelligence*, chapter 6, pages 89–102. Atlantis Press, 2012.
2. Kees Doets. *Basic model theory*. CSLI notes, 1996.
3. Robert Gilmore and Marc Lefranc. *The topology of chaos: Alice in stretch and squeezeland*. Wiley-VCH, 2011.
4. Maria Manzano. *Model theory*. Oxford, 1999.
5. Stephen Smale. Differentiable dynamical systems. *Bulletin of the American Mathematical Society*, 1967.
6. Tamás Tél and Márton Gruiz. *Chaotic dynamics: an Introduction based on classical mechanics*. Cambridge, 2006.
7. King Yin Yan. Wandering in the labyrinth of thinking – a cognitive architecture combining reinforcement learning and deep learning. to be submitted AGI 2017.